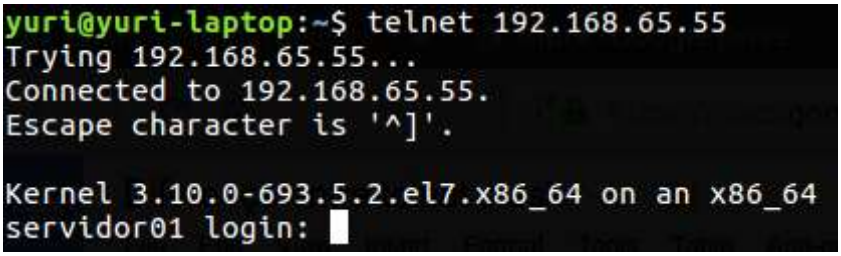
SSH, Telnet y las diferencias para conectar en un servidor

Yuri Matheus
23/12/2021



Vea la diferencia entre una simple **Telnet** y el **SSH** y como cada uno trata con las conexiones.

Estoy queriendo acceder un servidor para realizar algunas configuraciones que quedan en una sala de servidores junto con varios otros servidores.

Luego, si deseo configurar ese servidor, basta ir hasta la sala e introducirlo locamente. Pero ¿si no tuviera acceso a esa sala? O peor, ¿si esa sala estuviese en otro edificio del otro lado de la ciudad, o del estado, quien sabe hasta en ¿otro país? O ¿en un **cloud**?

La mejor manera es **acceder ese servidor remotamente**. De esa forma, no importa donde él está localizado, siempre conseguiremos acceder a él.

Existen diversos protocolos que consiguen realizar el acceso remoto, dos de los más conocidos son el **SSH** y el **Telnet**.

Con ambos protocolos nosotros conseguimos acceder a otro computador remotamente y crear un terminal virtual, que nada más es de que la emulación del terminal de la máquina accedida en nuestra propria máquina.

De esa manera, conseguimos ejecutar comandos, realizar configuraciones, o sea, tenemos total control del servidor mismo sin acceder a él físicamente.

Pero tu debes estarte preguntando:

> *"Por qué tener dos protocolos si ambos hacen la misma cosa?"*

Ambos protocolos pueden ser usados para acceder a computadores remotamente, pero existe algo muy importante que diferencia los dos. **La criptografía**.

Entendiendo la criptografía

Nosotros podemos definir **criptografía como la simple accion de tornar un mensage ilegíble**. A pesar de ser simple el concepto, existen diversos cálculos y métodos complejos usados para **garantizar que un mensaje** sea criptografía y que apenas los destinatarios sepan lo que significa.

Cuando un mensaje encriptado trafica por una red, como el Internet, cualquier uno que la intercepte consigue leer su contenido. O sea, si el contenido de ese mensaje contiene datos sigilosos, como claves, ellos quedan vulnerables.

Si ese mensaje está encriptada, el interceptador no conseguirá descubrir su contenido, pues apenas el receptor sabe como tornar el mensaje legible nuevamente.

Esa es la principal diferencia entre los protocolos **Telnet** y el **SSH**.

El Telnet no posee ninguna criptografia. El envia los mensajes en texto puro. De esa forma caso alguién intercepte esos datos conseguirá ver el contenido. ¿Pero como un hacker consigue interceptar esos mensajes?

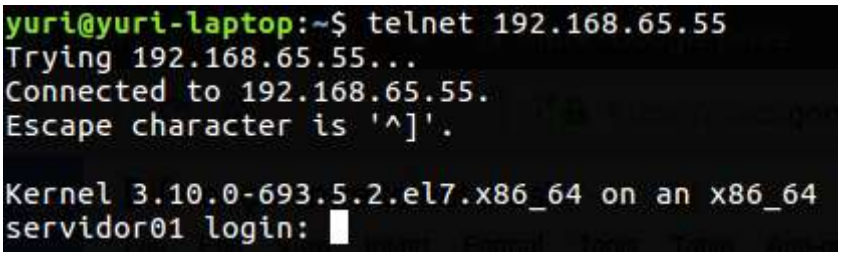
Existen diversas formas de interceptar un mensaje, ataques como **Man in the middle** son un ejemplo. Otra forma de interceptar esos mensajes es utilizar un olfateador (sniffer).

El sniffer queda olfateando paquetes en una red, o sea, el queda capturando los paquetes que pasan en una red. Los paquetes capturados pueden ser analizados y, si la comunicación no fuera encriptada, el consigue ver el contenido del mensaje. Un sniffer muy conocido es el **Wireshark**.

Olfateando el Telnet

Vamos hacer una conexión con el **Telnet** y usar el **Wireshark** para olfatear esa comunicación e ver si, de hecho, conseguimos capturar el contenido de los paquetes.

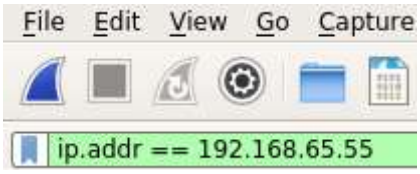
Entonces, en el terminal, podemos decir para el `telnet` conectarse a un host. En mi caso ese host es el `192.168.65.55`.



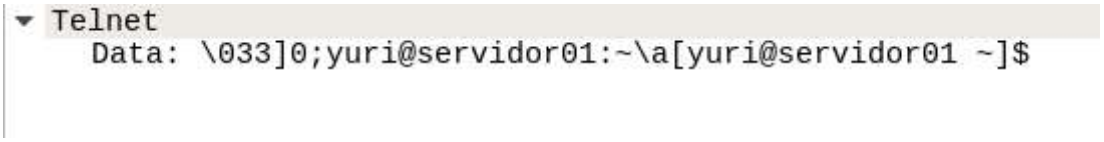
¡Digitamos el usuario y la clave y listo! Tenemos acceso al servidor.

Vamos a pedir ahora para el Wireshark comenzar a olfatear los paquetes que pasan por nuestra placa de red. Para quedar más fácil el análisis, podemos pedir para el filtrar los paquetes por la dirección IP de la máquina que estamos accediendo:

```
ip.addr == 192.168.65.55
```



Analizando los paquetes, conseguimos obtener algunas informaciones como quien es el usuario, en cual máquina el está logado y en cual pasta del sistema él se encuentra:



Además de eso, conseguimos capturar todo lo que es pasado por la red, como comandos, o parte de ellos:

```
▾ Telnet
    Data: > teste.txt
```

¿Será que utilizando el SSH tenemos acceso a esas informaciones?

SSH (Secure Socket Shell)

Vamos a conectar ahora utilizando el **SSH** y pedir para el **Wireshark** olfatear los paquetes.

Al contrario del `telnet` que pasamos apenas la dirección IP y después colocamos el usuario que deseamos acceder, con el `ssh` hablamos que deseamos acceder con el usuario en la direccion (@) y pasamos la dirección:

```
ssh yuri@192.168.65.55
```

```
yuri@yuri-laptop:~$ ssh yuri@192.168.65.55
yuri@192.168.65.55's password:
Last login: Thu Dec 14 10:00:38 2017 from ::ffff:192.168.65.111
[yuri@servidor01 ~]$
```

Analizando los paquetes, vemos que no tenemos acceso a las as informaciones:

```
▾ SSH Protocol
    Packet Length (encrypted): 47356444
    Encrypted Packet: a1a75543a496dca339c79afcd03e8621c3096d06e134b7ae...
```

Con el **SSH** mismo si el paquete fuera interceptado el contenido es ilegíble, pues el único que sabe como encriptar las informaciones y la máquina que estamos accesando.

Para saber más

Mismo el **SSH** encriptando los datos, todavia es posible descubrir el contenido de los paquetes. Existen varias técnicas, como el ataque Man in the Middle, que interceptan los paquetes e itentan abrir su contenido. El ataque que intentan combinaciones para desvendar la criptografía.

En los días de hoy, el **SSH** es mucho más usado de lo que el **Telnet** justamente por los beneficios de la seguridad.

Además de ser usado para [conectar remotamente, con el **SSH** nosotros también conseguimos hacer **transferencias seguras de archivos**.

Seguridad de redes es un asunto muy importante, principalmente en este mundo tan conectado. Entender como funcionan los protocolos, los equipamientos de redes, los tipos

de ataques, ayuda cuando estamos pensando en la política de seguridad.

Aquí en Alura tenemos una **formacion en redes**. En ella tu aprenderás sobre los protocolos y equipamientos de redes.

Yuri Matheus

Yuri es desarrollador e instructor. Es estudiante de Sistemas de Información en FIAP y graduado como Técnico en Informática en el Senac SP. Su enfoque es en las plataformas Java y Python y en otras áreas como Arquitectura de Software y Machine Learning. Yuri también actúa como editor de contenido en el blog de Alura, donde escribe, principalmente, sobre Redes, Docker, Linux, Java y Python..

Puedes leer también:

- Análisis de datos: analizando mi distribución con tres alternativas de visualización
- Formateo de moneda e internacionalización con Python
-
- Comprensión de listas en Python

ARTÍCULOS DE TECNOLOGÍA > DEVOPS

En Alura encontrarás variados cursos sobre DevOps. ¡Comienza ahora!